

GNU/Linux System Administration Commands

Ravi Kumar

October 11, 2005



Introduction

Monitoring the system

Process Management

File Systems and . . .

Collecting User Info

Synchronizing files . . .

Scheduling a job

Auditing in . . .

Finding Files

Managing Time

Communication Tools

Hardware Information

Backing up your . . .

Home Page

Title Page

◀◀

▶▶

◀

▶

Page 1 of 22

Go Back

Full Screen

Close

Quit

1. Introduction

This tutorial describes the various commands used for managing, modifying and monitoring a GNU/Linux based computer. Many people still believe that learning commands in GNU/Linux is difficult, or that only experts can master the commands in GNU/Linux so that it can be put to productive use. This tutorial is my endeavour to clearly document the main commands that can be used by an average GNU/Linux user so that he could maintain (administer?) his machine in a *relatively* secure and stable manner.

I had started this project with a view to know more about the various commands that can be used in GNU/Linux. I have taken great pains to see that this tutorial is free from errors. But then humans are prone to errors. And if you find anything out of context or any error in this tutorial, you are free to point it out to me. Also your suggestions on how to improve this tutorial are most welcome. You may also visit the site [All About Linux](#) to learn more about configuration and using of Linux. It covers all topics related to System and Network administration in Linux.

Contact me at: linuxhelp.ravi@gmail.com

2. Monitoring the system

GNU/Linux has a collection of tools for monitoring the system and collecting various system information like memory consumption, load on the

processor at any given time etc. I have listed the main utilities used for that purpose.

ps **tree** - Processes and parent-child relationships

top - Show all running processes

ps -auxw - Show Process Status

vmstat - Monitor Virtual Memory

free - Display amount of free and used memory in the system. This can also be done by **cat /proc/meminfo**.

cat /proc/sys/vm/freepages - Display the Virtual Memory “free pages”. One may increase or decrease this limit as follows :

```
echo 300 400 500 > /proc/sys/vm/freepages
```

cat /proc/filesystems - Display Filesystems currently in use.

uname -a - Print System Information

uptime - Tell how long the system has been running, number of users and system’s load average.

/sbin/lsmmod - List all currently loaded kernel modules.
Same as **cat /proc/modules**.

`/sbin/runlevel` - List the system's current runlevel.

`hostname` - Displays the system's hostname.

`service` - Display status of system services. For Eg:

`service --status-all`

`service --help`

3. Process Management

The basic monitoring commands such as `pstree` and `ps -auxw` and `top` will inform you of the processes running on your system. Sometimes a process must be terminated. To terminate a process, do the following -

1. Identify the Process. `pstree -p` OR `ps -auxw` OR `top`
2. Kill the Process. `kill <Process id number>` OR `killall <command line>`

This will perform an orderly shutdown of the process. If it hangs give a stronger signal as follows :

- `kill -9 <process id>`

A signal may be given to the process. The program must be programmed to handle the given signal. See `/usr/include/bits/signum.h` for a full list. For example, to restart a process after updating its configuration file, issue the command as follows :

- `kill -HUP <process id>`

In the above case, the HUP signal was sent to the process. The software was written to trap for the signal so that it could respond to it. If the software (command) is not written to respond to a particular signal, then the sending of the signal to the process is futile.

fuser -l - Identify all known signals

ulimit - Shell and process resources may be controlled and reported using the ulimit command. Display the limits of a shell using the bash command :

`ulimit -a`

Limits can be set for the number of open files and processes, memory and virtual memory etc.

4. File Systems and Storage Devices

4.1. Hard Drive Information

df -k - Report filesystem disk space usage. (-k reports in Kbytes).

du -sh - Calculates file space usage for a given directory. (and everything under it) (-s option summarizes)

mount - Displays all mounted devices, their mount point, filesystem, and access. Used with command line arguments to mount file system. Eg:
To mount a CDROM, do the following -
`mount -t iso9660 /dev/cdrom /mnt/cdrom`

showmount - Displays mount info for NFS filesystems.

cat /proc/swaps - Displays swap partition(s) size, type and quantity used.

cat /proc/ide/hda/any-file - Displays disk information held by kernel.

cat /proc/partitions - To view full list of disks and partitions that your system can see.

To add an extra harddrive, do the following :

1. Enter the drive into the partition table.

```
fdisk /dev/<drive>
```

2. Create file system.

```
mkfs -t ext2 /dev/<drive>
```

3. Mount the drive.

```
mount -t ext2 /dev/<drive> /<some suitable directory>
```

At this point one may optionally check the file system created with the command:

```
fsck /dev/<drive>
```

Note : **fsck** is NOT run against a mounted file system. Unmount it first if necessary.

cfdisk - Curses based disk partition table manipulator.

fsck - Check the filesystem command

IDE drives are referred to as hda for the first drive, hdb for the second etc ...IDE uses separate ribbon cables for primary and secondary drives. The partitions on each drive are referred numerically. The first partition on the first drive is referred to as hda1, the second as hda2, the third as hda3 ...and so on.

4.2. Linux IDE Naming conventions

Device	Description	Configuration
/dev/hda	1st (Primary) IDE controller	Master
/dev/hdb	1st (Primary) IDE controller	Slave
/dev/hdc	2nd (Secondary) IDE Controller	Master
/dev/hdd	2nd (Secondary) IDE Controller	Slave

[Introduction](#)[Monitoring the system](#)[Process Management](#)[File Systems and ...](#)[Collecting User Info](#)[Synchronizing files...](#)[Scheduling a job](#)[Auditing in...](#)[Finding Files](#)[Managing Time](#)[Communication Tools](#)[Hardware Information](#)[Backing up your ...](#)[Home Page](#)[Title Page](#)[<<](#)[>>](#)[<](#)[>](#)[Page 7 of 22](#)[Go Back](#)[Full Screen](#)[Close](#)[Quit](#)

Note : SCSI disks are labeled /dev/sda, sdb, etc ...

To make the drive a permanent member of your system and have it mount upon system boot, add it to your /etc/fstab file which holds all the file system information for your system. For more details **man fstab**.

5. Collecting User Info

who - Displays currently logged in users.

who -uH for idle time and terminal info.

w -s - Show who is logged on and what they are doing.

users - For all users logged in.

groups - Display groups you are part of.

groups <user-id> to display groups for a given user.

set - Display all environment variables in your current environment.

id - Display user and all group ids.

id <user-id> to display info for another user id.

last - Show listing of last logged in users.

history - Shell command to display previously entered commands.

[Introduction](#)[Monitoring the system](#)[Process Management](#)[File Systems and ...](#)[Collecting User Info](#)[Synchronizing files ...](#)[Scheduling a job](#)[Auditing in ...](#)[Finding Files](#)[Managing Time](#)[Communication Tools](#)[Hardware Information](#)[Backing up your ...](#)[Home Page](#)[Title Page](#)[<<](#)[>>](#)[<](#)[>](#)[Page 8 of 22](#)[Go Back](#)[Full Screen](#)[Close](#)[Quit](#)

5.1. Creating a new User Account

5.1.1. Command line method

useradd - Add a user to the system

useradd -G floppy - This will grant the user read/write privileges to the floppy (`/dev/fd0`) upon creation of user by adding user to group *floppy*.

userdel - Delete user from system. Purges user from `/etc/passwd`, `/etc/group` and `/etc/shadow` files.

userdel -r - Delete user and remove his home directory from the system. Other files will remain.

passwd - Assign a password to the user.

Note : Default directory configuration and files for a new user are copied from the directory `/etc/skel` .

The default shell is called **bash** (bsh) and is a cross of the UNIX **ksh** and **csh** command shells. The users personal bash shell customizations are held in `$HOME/.bashrc` file.

5.1.2. GUI method

linuxconf - This is a GUI frontend for creating and managing users in GNU/Linux.

5.1.3. File editing method

As **root**, edit files to add/remove a user as follows:

1. Create user entry in `/etc/passwd` file.
user:x:userid:group-id:user's full name:home directory:default shell
2. Create group in `/etc/group` file.
user:x:groupid
3. Create Home Directory
cd /home
mkdir user
4. Copy the default files
cp -pR /etc/skel/. /home/user
5. Change ownership of the files in `/home/user` to user
chown -R user.user /home/user

6. Creation of `/etc/shadow` and `/etc/gshadow` files require the execution of a program to encrypt passwords. Use `pwconv` and `grpconv`, to synchronize the shadow files.
 7. Create a password for the user.
`passwd user`
-

6. Synchronizing files on two computers: `rsync`

The command `rsync` helps the system administrator synchronize files on two separate computers. This is ideal for web site maintenance. Command to use `rsync` with `ssh`:

```
rsync -e ssh -p 777 --delete --include-from=files-to-sync.txt -avzu  
./ www.your-domain.com
```

Test with flags `-navzu`

File: `files-to-sync.txt`

```
+index.html  
-README  
+webpage-1.html  
+webpage-2.html
```

Introduction
Monitoring the system
Process Management
File Systems and ...
Collecting User Info
Synchronizing files...
Scheduling a job
Auditing in...
Finding Files
Managing Time
Communication Tools
Hardware Information
Backing up your ...

[Home Page](#)[Title Page](#)[<<](#)[>>](#)[<](#)[>](#)

Page 11 of 22

[Go Back](#)[Full Screen](#)[Close](#)[Quit](#)

+webpage-3.html

Files to include (+) and files which are excluded from synchronization (-).

7. Scheduling a job

7.1. Scheduling a re-occurring task : `cron`

```
crontab -e # edit new/ existing cron job
crontab -l # list the cron jobs
crontab -r # remove the cron jobs (man crontab for details)
To know more about cron,
man cron,
man crontab
```

7.2. Scheduling a task using : `at`

The `at` command will schedule *single jobs*. (`cron` is for re-occurring jobs)
The daemon `/usr/sbin/atd` will run jobs scheduled with the `at` command.
Access control to the command is controlled using the files `/etc/at.allow`
(list of user id's permitted to use the `at` command) and `/etc/at.deny`.
Time is specified before the date in the following formats :

- at HH:MM month-name day with an optional year
- at midnight MMDDYY
- at HH PM today
- at noon DD.MM.YY
- at 14:30 19.03.06
- at noon tomorrow

The **at** command will respond with its "**at>**" prompt upon which you enter the command you wish to execute followed by "**Enter**". More commands may be entered. When done enter "**control-d**".

The **at** commands are as follows :

at - schedule job at a specified time

atq - lists the user's pending jobs or all if root

atrm - deletes jobs, identified by their job number

batch - executes commands when system load levels permit (based on a specified system load)

atrun - for backward compatibility. Functionality built into **at**.

Introduction
Monitoring the system
Process Management
File Systems and ...
Collecting User Info
Synchronizing files...
Scheduling a job
Auditing in ...
Finding Files
Managing Time
Communication Tools
Hardware Information
Backing up your ...

[Home Page](#)

[Title Page](#)

◀◀

▶▶

◀

▶

Page 13 of 22

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)

8. Auditing in GNU/Linux

8.1. System Log files

`/var/log/messages` - System log messages

`/var/log/secure` - Logging by PAM of network access attempts.

`/var/log/dmesg` - Log of system boot. Also see command `dmesg`.

`/var/log/boot.log` - Log of System *init* processes.

`/var/log/lastlog` - Requires the use of the `lastlog` command to examine its contents. This command prints time stamp of the last login of system users.

`var/log/maillog` - Log from sendmail daemon.

8.2. Rotate Log files : `logrotate`

Many system and server application programs such as Apache, generate log files. If left unchecked they would grow large enough to burden the system and application. The `logrotate` program will periodically backup the log file by renaming it. The program will also allow the system administrator to set the limit for the number of logs or their size. There is also the option to compress the backed up files.

`/etc/logrotate.conf` - Configuration file for logrotate.

`/etc/logrotate.d/` - Directory for logrotate configuration scripts.

9. Finding Files

9.1. Using the find command

This is a very good program for finding files in GNU/Linux. To know more about `find`, `man find`. Given below are a few examples which illustrate the use of `find` command.

```
find ./ -name "*.conf" -print
```

Find all files of a given type from current directory on down.

```
find / -user 501 -print
```

Find all files owned by a user (defined by user id number), on the system.

```
find /home -size +50000000c -print
```

Find all user files larger than 5Mb.

```
find / -perm -0002 -type d -print
```

Find all world writable directories.

```
find / -perm -0002 -type f -print
```

Find all world writable files.

```
find / -mtime 2 -o -ctime 2
```

Find files modified in the last two days.

```
find / -nouser -o -nogroup -print
```

Find files with no user.

Note: You can get a wonderful tutorial on using `find` at soft-lab.ece.ntua.gr

9.2. Other ways of finding files

locate / **slocate** - Find location/list of files which contain a given partial name.

which - Find executable file location of a command. The command must reside in the path.

whereis - Find executable file location of a *command* and *related files*.

rpm -qf file - Display name of RPM package from which the file was installed.

10. Managing Time

10.1. Setting System Clock using : `date`

`date MMDDhhmmYYYY` - This is the format in which you set the date and time on your machine. Here,

MM - Month (Eg: 1 for jan, 2 for Feb ...)

DD - Date

hh - Hour

mm - Minutes

YYYY - Year (Eg: 2004)

For example, to set the date and time as Jan 23rd 2004, 1.40PM, you give the command :

```
date 012313402004
```

10.2. Setting time from a remote server : rdate

rdate -p hostname - Print the time returned by the remote host.

rdate -s hostname - Set the system time to the returned time.

For example try `rdate -p time.ucla.edu`

Typically many web servers set their time to GMT due to the worldwide nature of their service. Internally UNIX systems use Coordinated Universal Time (UTC) which is the number of seconds since Jan 1, 1970 0 hrs. "Calendar Time" is then calculated based on your time zone and whether you are on Standard or Daylight Savings time.

10.3. Network Time Protocol

The daemon `ntpd` will continually monitor time. The init script, `/etc/rc.d/init.d/ntpd` issues the command `/usr/sbin/ntpdate` to set the time. It expects time servers to be listed in `/etc/ntp/step-tickers`. You may also issue the command:

```
ntpdate -b -s time.nist.gov ns.arc.nasa.gov tick.usno.navy.mil
```

This will hang without an internet connection.

Time servers

- time.nist.gov
- ns.arc.nasa.gov
- tick.usno.navy.mil

11. Communication Tools

11.1. Checking your mail : `mail`

” You have new Mail”

When you login, this message may greet you. The system will often send a mail message to the root user after the completion of some cron jobs, software installation or as an error message meant to alert the system administrator. Type the console command `mail` to view your incoming

mail. The following simple commands will help you navigate through this simple mail client.

Mail Command	Description
?	Lists command Help
h	Print Mail Headers
h 1	Print mail headers starting with message number 1
q	Quit
t	Type current message
t 4	Type out message 4 to the console.
n	Type out next message
d	Delete the active message
d 4	Delete message number 4
d 1-4	Delete messages numbered 1 to 4.

11.2. Miscellaneous tools for Communication

talk - This is a visual communication program which copies lines from your terminal to that of another user.

```
talk <user@machine name> <ttyname>
```

write - This program allows you to communicate to other users by copying lines from your terminal into their's.

```
write user [tty name]
```

mesg - Control write access to your terminal by other users.
mesg [y|n]

12. Hardware Information

/sbin/lspci - List all PCI devices (Result of probe). Also try
lspci -vvx and **cat /proc/pci**

cat /proc/interrupts - List IRQ's used by system.

cat /proc/ioports - List I/O ports used by system.

cat /proc/dma - List DMA channels and device used by system.

cat /proc/cpuinfo - List info about CPU

13. Backing up your system

13.1. Tape ARchive : **tar**

The command **tar** is used to create simple backups on hard disks, magnetic tapes and so on. Following are a few examples of the use of **tar** command :

tar -cvf /dev/st0 /home /opt - This will backup the files, directories and all it's subdirectories and files of the directories **/home** and **/opt** to the first SCSI tape device. (**/dev/st0**)

tar -xvf /dev/st0 - This restores files from backup SCSI tape device.

tar -z -cvf /dev/st0 /home /opt - Backup using compression to put more data on tape.

tar -tf /dev/st0 - List contents of tape.

tar -tzf /dev/st0 - List contents of compressed backup tape.

tar -cvf /dev/fd0 /home/user1 - Backup directory to a floppy. When restored it requires *root* because the root of the backup is **/home**.

tar -cvf /dev/fd0 src - Backup sub-directory to floppy using a relative path.

tar -cvf /dev/fd0 -C /home/user1 src - Backup sub-directory to floppy using a defined relative path.

tar -xvf /dev/fd0 - Restore from floppy.

tar -z -cvf backup-file.tar.gz -C /home/user1/src project-x - Backup directory to a compressed archive file.

tar -tzf backup-file.tar.gz - List contents of the compressed archive.

tar -xzf backup-file.tar.gz - Restore the files from the compressed archive file.

13.2. Other ways of doing a backup

rsync - This command can be used to do a snap-shot style backup in GNU/Linux. For more info, read the article - [Easy Backups using rsync and GNU/Linux](#).

gzip - Strictly speaking, this is not a backup program, but a file compression program which can be used in combination with **tar** to create compressed archives.

Also visit [linux-backup.net](#) for a wealth of information on how to go about taking a backup of your files in GNU/Linux.
